

Incloure el codi Javascript en la nostra web	1
Sentències	1
Per fer comentaris en javascript	2
Comentari multilínea	2
Interacció amb el navegador	2
Mostrant missatge per consola	2
Mostrant finestra amb missatge	2
Mostrant missatge amb html <code>document.write('frase' + '&lt;br&gt;&lt;br&gt;');</code>	2
Mostrant missatge directament al html - DOM	2
Variables	3
Estilos para nombrar variables con más de una palabra	4
Scope de les variables	4
Scope Local	4
Scope Global	5
Assignacions	5
Tipus de dades	6
Operadors	7
Operadors aritmètics	7
Operadors lògics	9
Exercicis	10
Sintaxi bàsica	10
Referències	10

## Incloure el codi Javascript en la nostra web

Mirar píldora [HTML\\_estructura\\_basica\\_web](#)

## Sentències

En un llenguatge de programació, com JavaScript, cadascuna de les instruccions que formen el programa que executarà l'ordinador s'anomenen sentències.

Com es veurà a continuació, les sentències en JavaScript estan formades per comentaris, variables, operadors, condicionals, bucles i paraules clau.

Com a bona pràctica, encara que no és necessari, separem les sentències amb el punt i coma “;”. També és recomanable utilitzar un únic punt i coma “;” per línia, tot i que Javascript permet declarar-ne varis en una mateixa línia.

## Per fer comentaris en javascript

Poseu dues barres // al començament de la línia o just davant del codi a comentar

```
// Este un comentario de una única línea  
alert("Escibiendo comentarios en javascript!"); //Aquí puedo poner una nota de  
lo que hace esta línea
```

## Comentari multilínea

/\* al principi i \*/ al final del comentari, pots incloure totes les línies que vulguis.

```
/*  
    alert("Esto no se ejecuta");  
    alert("Y esto tampoco");  
    Y este texto puede decir lo que yo quiera  
    para acordarme de algo  
*/
```

## Interacció amb el navegador

Mostrant missatge per consola

```
Console.log("Missatge")
```

Mostrant finestra amb missatge

```
Window.alert("Missatge") // El mètode "alert" mostra una finestra amb el missatge;
```

Mostrant missatge amb html

```
document.write('frase' + '<br><br>');
```

Mostrant missatge directament al html - DOM

```
<!DOCTYPE html>  
<html lang="en">  
  
<head>  
    <meta charset="UTF-8">  
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Document</title>
</head>

<body>
  <h2>JavaScript new Date()</h2>
  <p id="demo"></p>
  <script>
    const d = new Date();
    document.getElementById("demo").innerHTML = d;
  </script>
</body>
</html>
```

## Variables

A JavaScript, s'ha de tenir en compte certes normes i consideracions sobre les variables:

Amb Vanilla Javascript una variable es declara amb la paraula reservada «**var**»

### **i** OJO, amb ES6

*Es substitueix **let** per **var** alhora de declarar variables*

***Apareix **const** per declarar variables amb valor constant. Exemple:***

*const PI = 3.141592653589793;*

*PI = 3.14; // This will give an error*

*PI = PI + 10; // This will also give an error*

Exemple:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Exemple declaració variables</title>
  </head>
  <body>
    <script>
      //declaració de la variable
      var a;
      //també es pot declarar més d'una variable
      //a la mateixa línia
      var b, c, d, e, f, g;
      //fins i tot com s'ha vist anteriorment
```

```
        //amb v ries sent ncies en una l nia
        var h; var i; var j;
    </script>
</body>
</html>
```

- Els identificadors de les variables:
  - Han de ser  nics,  s a dir, no es poden repetir.
  - Han de comen ar amb:
    - una lletra,
    - el s mbol de subratllat “\_”,
    - o el de d lar “\$”.
- Nom s poden contenir:
  - lletres,
  - n meros, i
  - els s mbols “\_” i “\$”.
- Tampoc es poden utilitzar paraules reservades per a declarar identificadors de variables (“while”, “for”, “new”...). La llista sencera es pot consultar, per exemple, a l’associaci  que defineix l’est ndard. A continuaci  est  enlla ada l’est ndard d’aquest any: <https://www.ecma-international.org/ecma-262/>.
- Finalment, cal tenir en compte que a JavaScript:
  - Es distingeix entre maj scules i min scules. No  s el mateix la variable “a” que “A”.
  - Les variables s n din miques (per exemple, poden canviar de tipus ‘number’ a ‘string’, no com a altres llenguatges).

## Estilos para nombrar variables con m s de una palabra

```
var nombreProducto = 'Monitor 30 Pulgadas'; // CamelCase
var nombre_producto = 'Monitor 30 Pulgadas'; //underscore
var NombreProducto = 'Monitor 30 Pulgadas'; // pascal case
var nombreproducto = 'Monitor 30 Pulgadas';
```

## Scope de les variables

Entenem per scope cadascuna de les zones que limiten l’acc s a les dades dins del nostre codi.  s la manera de posar l mits a quin codi veu qu  quan s’executa.

### Scope Local

Quan pots accedir a una variable nom s en certa part del codi, es diu que aquesta variable est  declarada al **scope local**. Aquestes tamb  s n conegudes com a **variables locals**. Un exemple  s quan una variable est  declarada dins d’un bloc o una funci . Si intenteu accedir

a aquesta variable fora d'aquesta funció o bloc, tindreu un error que dirà que la variable no està definida.

Imaginem aquest cas:

```
function funcionQueDefineUnScopeLocal() {  
  var nemesis = "Robotnik";  
  console.log(nemesis); // Robotnik  
}
```

## Scope Global

Es diu que una variable té **scope global** quan està declarada fora d'una funció o d'un bloc. També són conegudes com a **variables globals**. Podeu accedir a aquest tipus de variables des de qualsevol part del vostre codi, ja sigui dins o fora d'una funció.

```
var personaje = 'Sonic';
```

```
function funcionQueDefineUnScopeLocal() {  
  var nemesis = "Robotnik";  
  console.log(personaje); // Sonic  
  console.log(nemesis); // Robotnik  
}  
console.log(personaje); // Sonic  
console.log(nemesis); // undefined
```

**i** Tot i que JavaScript ens permet declarar una variable com a global, no és una bona pràctica. Una de les raons és perquè tenim la possibilitat de declarar dues variables globals a diferents parts del codi amb el mateix nom sense notar-ho.

**i** Des d'ES6 qualsevol parell de claudàtors {} defineixen un scope, per la qual cosa a partir d'ara disposem del ja existent functional scope però també de lexical scope. Ho veurem més endavant

## Assignacions

Les assignacions de valors a variables es fan amb l'operador "=" i tenen lloc de dreta a esquerra.

```
<!DOCTYPE html>  
<html>  
  <head>
```

```
    <title>Exemple assignacions</title>
  </head>
  <body>
    <script>
      //declaració de la variable
      var a;
      //assignació del valor 17
      a=17;
      //declaració i assignació de la variable
      var b=17;
      //també es pot assignar un valor des de la pantalla
      var c=prompt("Introdueix un valor");
      console.log(c);
      alert(c);
    </script>
  </body>
</html>
```

## Tipus de dades

JavaScript distingeix entre els següents tipus de dades bàsiques:

- Number  
`let n = 123;`  
`n = 12.345;`
- Strings  
`let str = "Hola";`
- Boolean  
`let isGreater = 4 > 1;`  
`alert(isGreater); //True (el resultado de la comparación es "sí")`
- "null"  
Es sólo un valor especial que representa "nada", "vacío" o "valor desconocido".  
`let age = null;`
- "undefined"  
El significado de undefined es "valor no asignado". Si una variable es declarada pero no asignada, entonces su valor es undefined:  
`let age;`  
`alert(age); // muestra "undefined"`

**i** *Nota: Per saber el tipus de variable fem servir typeof*  
*typeof age // "number"*  
*typeof isGreater // boolean*

Els següents valors no s'entenen com Tipus de dades, però els podem obtenir en el nostres programes:

- NaN

Cuando intentamos realizar operaciones que no pueden llevarse a cabo

```
var n1 = 'Texto';  
var n2 = 3;  
var result = n1 * n2;  
console.log('El resultado es: '+result )
```

En el código anterior recibimos el valor NaN debido a que intentamos multiplicar un valor en texto con un valor numérico.

**i** *Nota: Per validar si una variable es NAN fem servir: isNaN()*

- Infinity

Obtendremos un valor Infinity en JavaScript cuando se excede el límite superior (+) o inferior (-) de coma flotante que son: 1.797693134862315E+308. y - 1.797693134862315E+308.

```
console.log(1.7976931348923157E+10309)
```

**i** *Nota: Per validar si una variable es Infinity fem servir: isFinite()*

## Operadors

Hi han dos tipus d'operadors de variables:

- Aritmètics
- Lògics

### Operadors aritmètics

Els operadors aritmètics s'utilitzen amb variables de tipus Number o String. Es poden diferenciar les següents operacions:

- |                 |         |         |
|-----------------|---------|---------|
| • Suma          | a=b+c;  |         |
| • Resta         | a=b-c;  |         |
| • Multiplicació | a=b*c;  |         |
| • Exponent      | a=b**c; | //a=b^c |
| • Divisió       | a=b/c;  |         |
| • Mòdul         | a=b%c   |         |

- Increment `a++;`
- Decrement `a--;`

Exemple:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Exemple Operadors Aritmètics</title>
  </head>
  <body>
    <script>
      //Operadors aritmètics
      var a, b, c;
      console.log("Declaració de variables");
      console.log("a = "+a+"");
      console.log("b = "+b+"");
      console.log("c = "+c+"");
      //assignació de valors
      console.log("Assignació de valors");
      a=0, b=10, c=20;
      console.log("a = "+a+"");
      console.log("b = "+b+"");
      console.log("c = "+c+"");
      //suma
      console.log("Operador suma '+'");
      a=b+c;
      console.log("a = b + c = "+a+"");
      //resta
      console.log("Operador resta '-'");
      a=b-c;
      console.log("a = b - c = "+a+"");
      //multiplicació
      console.log("Operador multiplicació '*'");
      a=b*c;
      console.log("a = b * c = "+a+"");
      //exponent
      console.log("Operador multiplicació '*'");
      a=b**c;
      console.log("a = b * c = "+a+"");
      //divisió
      console.log("Operador divisió '/'");
      a=b/c;
      console.log("a = b / c = "+a+"");
```



```

        //mòdul
        console.log("Operador mòdul '%');
        a=b%c;
        console.log("a = b % c = "+a+");
        //increment
        console.log("Operador increment '++');
        b++;
        console.log("El resultat de 'b++;' és "+b+");
        //decrement
        console.log("Operador decrement '--');
        b--;
        console.log("El resultat de 'b--;' és "+b+");
    </script>
</body>
<html>

```

Alternativament, alguns operadors aritmètics poden incloure l'operador d'assignació:

- |                 |       |           |
|-----------------|-------|-----------|
| • Suma          | a+=b; | // a=a+b; |
| • Resta         | a-=b; | // a=a-b; |
| • Multiplicació | a*=b; | // a=a*b; |
| • Divisió       | a/=b; | // a=a/b; |
| • Mòdul         | a%=b; | // a=a%b; |

## Operadors lògics

Els operadors lògics retornen una variable de tipus boolean en funció de la comparació.

- |                                   |     |
|-----------------------------------|-----|
| • Igual valor                     | ==  |
| • Igual valor i tipus             | === |
| • Diferent valor                  | !=  |
| • Diferent valor o diferent tipus | !== |
| • Més gran                        | >   |
| • Més petit                       | <   |
| • Més gran o igual                | >=  |
| • Més petit o igual               | <=  |
| • Conjunció lògica                | &&  |
| • Disjunció lògica                |     |
| • Negació lògica                  | !   |
| • Operador ternari                | ?   |

Els operadors lògics normalment els farem servir amb condicionals, que veurem en una pílhora posterior. Fem un tastet d'un condicional fent servir operadors lògics

```

if (condition) {
    // block of code to be executed if the condition is true
} else {

```

```
// block of code to be executed if the condition is false  
}
```

Per practicar una mica amb els operadors lògics:

[https://www.w3schools.com/js/js\\_comparisons.asp](https://www.w3schools.com/js/js_comparisons.asp)

## Exercicis

### Sintaxi bàsica

1. Declara dues variable numèriques i assigna dos números, Suma'ls i mostra el valor per consola.
2. Utilitzant tant les cometes simples com les dobles per a inicialitzar la variable, crea un programa que mostri en una finestra:  
    I'm = I am  
    You're = You are
3. Implementa un programa que comprovi que les dades introduïdes són números.
4. Fes un programa que demani dos números a l'usuari i informi per pantalla de quin és el menor i el major, o si són iguals.
5. Fes un programa que demani un número a l'usuari i calculi si és parell o senar.
6. Fes un programa que demani a l'usuari l'altura i el pes i mostri per pantalla el seu índex de massa corporal (IMC)
7. Considerant la següent assignació: `let marca="total"+ 16 +4`  
    obté per consola total20

## Referencies

[https://www.w3schools.com/js/js\\_syntax.asp](https://www.w3schools.com/js/js_syntax.asp)  
[https://www.w3schools.com/js/js\\_comments.asp](https://www.w3schools.com/js/js_comments.asp)  
[https://www.w3schools.com/js/js\\_variables.asp](https://www.w3schools.com/js/js_variables.asp)  
[https://www.w3schools.com/js/js\\_let.asp](https://www.w3schools.com/js/js_let.asp)  
[https://www.w3schools.com/js/js\\_const.asp](https://www.w3schools.com/js/js_const.asp)  
[https://www.w3schools.com/js/js\\_operators.asp](https://www.w3schools.com/js/js_operators.asp)  
[https://www.w3schools.com/js/js\\_arithmetic.asp](https://www.w3schools.com/js/js_arithmetic.asp)  
[https://www.w3schools.com/js/js\\_assignment.asp](https://www.w3schools.com/js/js_assignment.asp)  
[https://www.w3schools.com/js/js\\_datatypes.asp](https://www.w3schools.com/js/js_datatypes.asp)

Javascript

Sentències\_tipus de variables i operadors i scope.

v1.0

M01: Desenvolupament Web

Curs: Front End Web Developer